

## Challenge Type: Forensics

### Difficulty: Easy

For this challenge, we are given a file called “recdisk.img”. We have to extract the flag from this.

As a part of my preliminary checks for Forensic challenges, I do the below

1. Use the *strings* command to see if the flag is present as plaintext in the file.
2. Use [Binwalk](#) to check for embedded files.

Applying those on recdisk.img, we get to know that

1. There is a file called *flag.png* in it.
2. There are 2 png images embedded( indices 92848 and 486064).

```
root@kali:~/Downloads/tamuctf# strings recdisk.img | grep flag
flag.png
root@kali:~/Downloads/tamuctf# binwalk recdisk.img
```

DECIMAL	HEXADECIMAL	DESCRIPTION
76464	0x12AB0	PDF document, version: "1.4"
76535	0x12AF7	Zlib compressed data, default compression
76774	0x12BE6	Zlib compressed data, default compression
87882	0x1574A	Zlib compressed data, default compression
92848	0x16AB0	PNG image, 329 x 17, 8-bit grayscale, non-interlaced
273072	0x42AB0	Zip archive data, at least v2.0 to extract, name: _rels/.rels
273346	0x42BC2	Zip archive data, at least v2.0 to extract, name: word/settings.xml
273595	0x42CBB	Zip archive data, at least v2.0 to extract, name: word/_rels/document.xml.rels
273971	0x42E33	Zip archive data, at least v2.0 to extract, name: word/fontTable.xml
274385	0x42FD1	Zip archive data, at least v2.0 to extract, name: word/numbering.xml
275259	0x4333B	Zip archive data, at least v2.0 to extract, name: word/media/image1.jpeg
375699	0x5BB93	Zip archive data, at least v2.0 to extract, name: word/charts/chart1.xml
376594	0x5BF12	Zip archive data, at least v2.0 to extract, name: word/styles.xml
377822	0x5C3DE	Zip archive data, at least v2.0 to extract, name: word/document.xml
382592	0x5D680	Zip archive data, at least v2.0 to extract, name: docProps/app.xml
382823	0x5D767	Zip archive data, at least v2.0 to extract, name: docProps/core.xml
383169	0x5D8C1	Zip archive data, at least v2.0 to extract, name: [Content_Types].xml
384353	0x5DD61	End of Zip archive, footer length: 22
387760	0x5EAB0	JPEG image data, EXIF standard
387772	0x5EABC	TIFF image data, little-endian offset of first image directory: 8
486064	0x76AB0	PNG image, 1068 x 966, 8-bit/color RGBA, non-interlaced
486128	0x76AF0	Zlib compressed data, best compression
500566	0x7A356	Zlib compressed data, default compression

Now, its very likely that the flag is in one of these 2 png files. So we extract them using the *dd* command and name them *flag1.png* and *flag2.png*.

```
root@kali:~/Downloads/tamuctf# dd if=recdisk.img of=flag1.png bs=1 count=64 skip=486064
64+0 records in
64+0 records out
64 bytes copied, 0.00288294 s, 22.2 kB/s
root@kali:~/Downloads/tamuctf# ls -le
flag1.png  recdisk.img
root@kali:~/Downloads/tamuctf# strings flag1.png
IHDR
8dzTXtRaw profile type exif
root@kali:~/Downloads/tamuctf# dd if=recdisk.img of=flag2.png bs=1 count=180224 skip=92848
180224+0 records in
180224+0 records out
180224 bytes (180 kB, 176 KiB) copied, 0.926538 s, 195 kB/s
root@kali:~/Downloads/tamuctf# strings flag2.png
```

I first thought of using *strings* again to see if that got me the flag. It did not but I was able to see it after opening *flag2.png*.

**Flag:** gigem{wh3r3\_w3r3\_601n6\_w3\_d0n7\_n33d\_h34d3r5}